CSE422

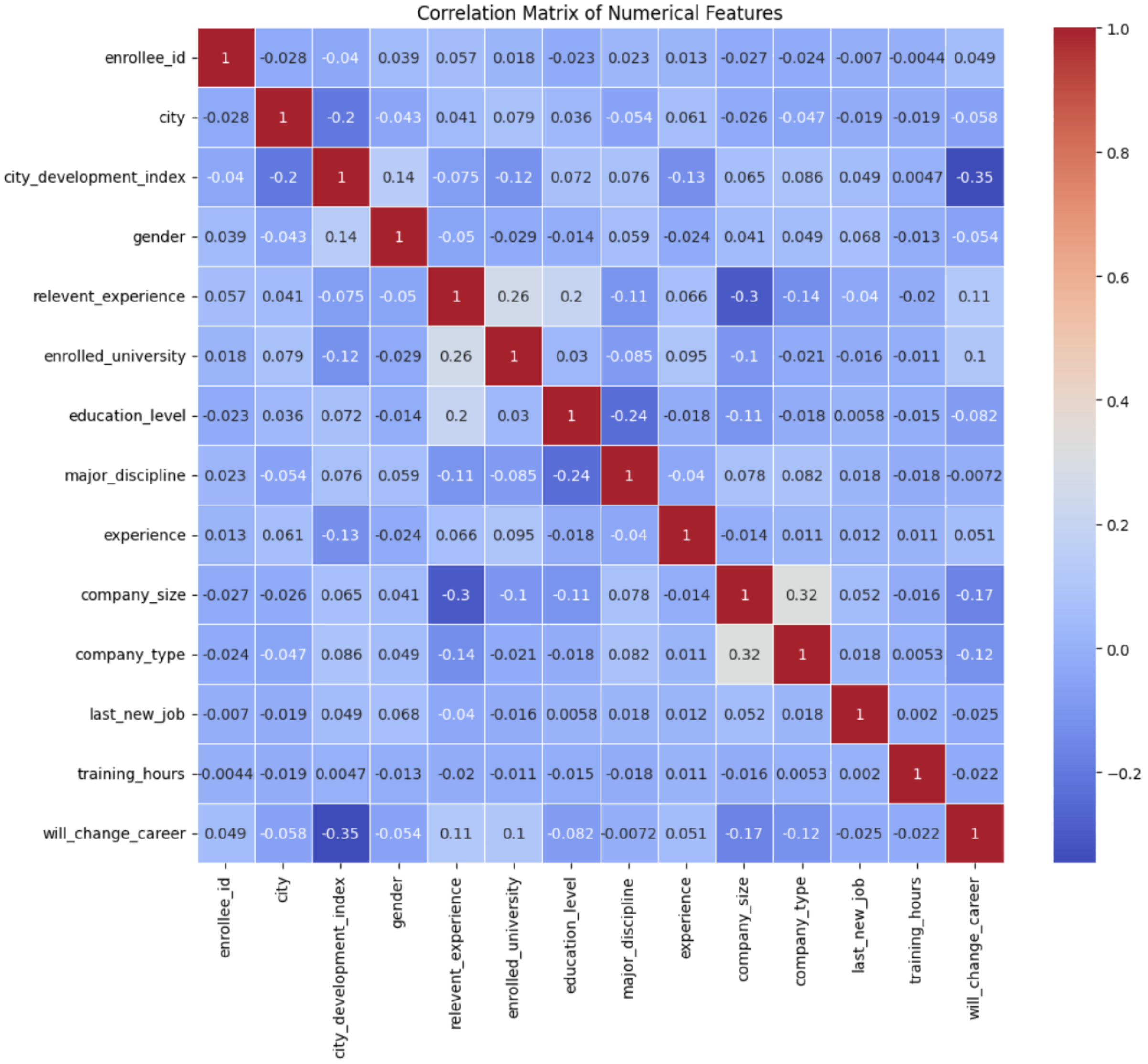
ARTIFICIAL INTELLIGENCE

Table of Content

1.Introduction: In today's rapidly evolving job market, career switching has become increasingly common due to changing interests, economic shifts, or industry demands. Our project focuses on predicting the likelihood of an individual switching careers based on various factors such as enrolled ID, city, city development index, gender, relevant experience, enrolled university, education level, major discipline, experience, company size, company type, last new job, training hours, will change career. The aim is to analyse personal and professional attributes to assess the risk or probability of a career change, which can aid both individuals and organizations in making informed decisions. The motivation behind this project is to apply machine learning techniques to understand the patterns and key indicators that lead to a career switch, ultimately helping career counsellors, HR departments, and job seekers proactively address professional development and career planning.

2. Dataset Description: This dataset presents a classification problem where the target variable "will change career" consists of discrete class labels indicating whether an individual is likely to switch careers or not. The dataset contains 14 features and about 7000 data points from there few of them are categorical features among these, several features are categorical, such as city , gender, relevant experience, enrolled university, education level, major discipline, company size, company type, and last new job. Additionally, there are numerical and qualitative features like enrolled ID , city development index, training hours, and years of experience. One of the key considerations in working with this dataset is the potential class imbalance in the target variable, which needs to be addressed to ensure robust model performance.

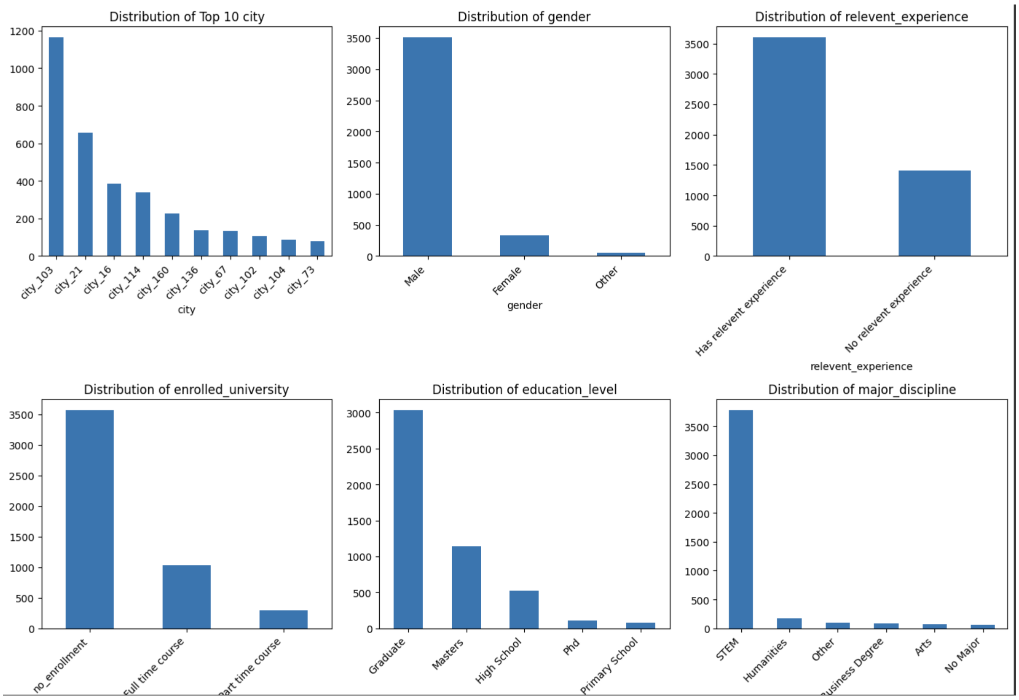
Correlation:



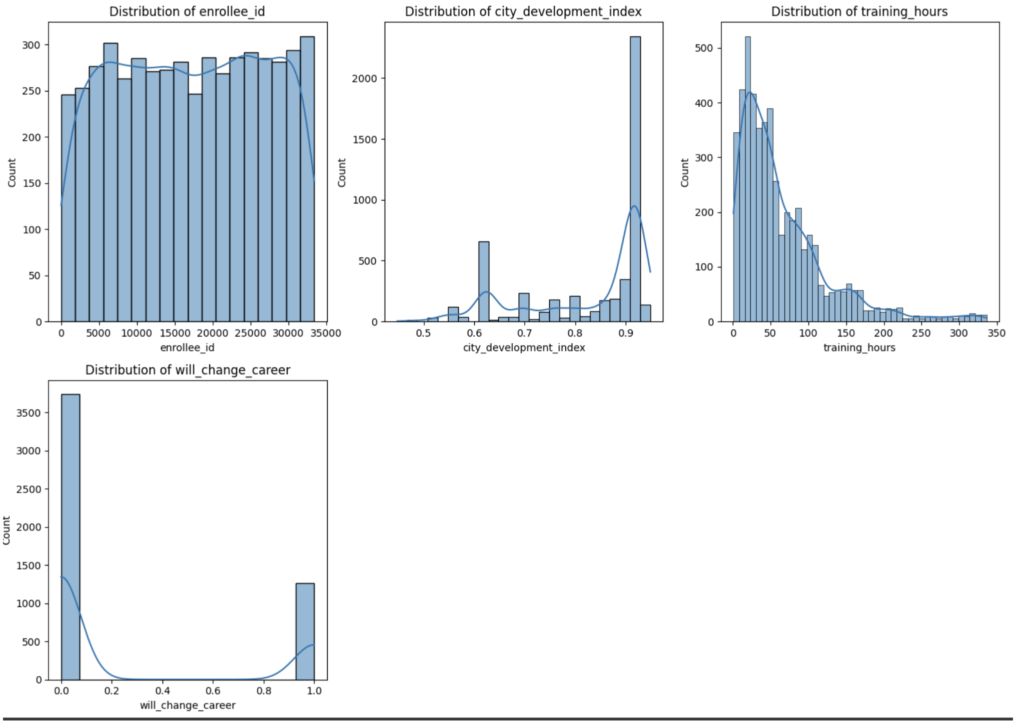
Correlation Matrix of Numerical Features

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	company_size	company_type	last_new_job	training_hours	will_change_career
enrollee_id	1	-0.028	-0.04	0.039	0.057	0.018	-0.023	0.023	0.013	-0.027	-0.024	-0.007	-0.0044	0.049
city	-0.028	1	-0.2	-0.043	0.041	0.079	0.036	-0.054	0.061	-0.026	-0.047	-0.019	-0.019	-0.058
city_development_index	-0.04	-0.2	1	0.14	-0.075	-0.12	0.072	0.076	-0.13	0.065	0.086	0.049	0.0047	-0.35
gender	0.039	-0.043	0.14	1	-0.05	-0.029	-0.014	0.059	-0.024	0.041	0.049	0.068	-0.013	-0.054
relevent_experience	0.057	0.041	-0.075	-0.05	1	0.26	0.2	-0.11	0.066	-0.3	-0.14	-0.04	-0.02	0.11
enrolled_university	0.018	0.079	-0.12	-0.029	0.26	1	0.03	-0.085	0.095	-0.1	-0.021	-0.016	-0.011	0.1
education_level	-0.023	0.036	0.072	-0.014	0.2	0.03	1	-0.24	-0.018	-0.11	-0.018	0.0058	-0.015	-0.082
major_discipline	0.023	-0.054	0.076	0.059	-0.11	-0.085	-0.24	1	-0.04	0.078	0.082	0.018	-0.018	-0.0072
experience	0.013	0.061	-0.13	-0.024	0.066	0.095	-0.018	-0.04	1	-0.014	0.011	0.012	0.011	0.051
company_size	-0.027	-0.026	0.065	0.041	-0.3	-0.1	-0.11	0.078	-0.014	1	0.32	0.052	-0.016	-0.17
company_type	-0.024	-0.047	0.086	0.049	-0.14	-0.021	-0.018	0.082	0.011	0.32	1	0.018	0.0053	-0.12
last_new_job	-0.007	-0.019	0.049	0.068	-0.04	-0.016	0.0058	0.018	0.012	0.052	0.018	1	0.002	-0.025
training_hours	-0.0044	-0.019	0.0047	-0.013	-0.02	-0.011	-0.015	-0.018	0.011	-0.016	0.0053	0.002	1	-0.022
will_change_career	0.049	-0.058	-0.35	-0.054	0.11	0.1	-0.082	-0.0072	0.051	-0.17	-0.12	-0.025	-0.022	1

Plot Distribution of Categorical Values



Plot Distribution of Numerical Values



3. **Dataset Pre-processing:**

It is important to inspect the dataset before training machine learning models to avoid some potential issues. In the dataset there might be some null values. Now from the dataset we have separate the features and target value or label. By runnig this command df.isnull( ) we saw that there are some null values in our dataset. To minimize those null values, for categorical features we replaced the value with the mode (most frequent value) and for numerical features we replaced the value with median. We also noticed that there are some dtype = object values that means we got categorical features in our dataset. Machine learning algorithm such as logistic regression and KNN require numerical input. Therefore, categorical variables must be transformed into numerical formats to ensure compatibility with the models. To solve this problem, we used encoding to convert categorical features into numerical format. This creates new binary columns for each category in a categorical feature.

Feature Scaling: We applied feature scaling to ensure that all features have a comparable scale, which is essential for distance-based models like KNN and gradient-based methods like Logistic Regression. We used Standardization to scale the features. This method transforms the data to have a mean of 0 and a standard deviation of 1.

4. Dataset Splitting: The dataset was split into training and testing sets. We used an 70-30 split, where 70% of the data was used for training and 30% for testing. This was implemented using the train_test_split function from sklearn. We also used random start and stratify parameters to improve the reliability of the model evaluation.

5. Model Training and Testing: Three models were trained and tested: Logistic Regression, K-Nearest Neighbors (KNN), and Decision Tree. We implemented these models to compare the performance of these models on a multiclass classification problem and determine the best-performing one.

- **K-Nearest Neighbors (KNN):** A distance-based algorithm where predictions are made based on the closest k neighbors in the feature space.
- **Decision Tree:** A tree-based algorithm that splits data based on feature values to make predictions.
- **Logistic Regression:** A linear model for classification problems.

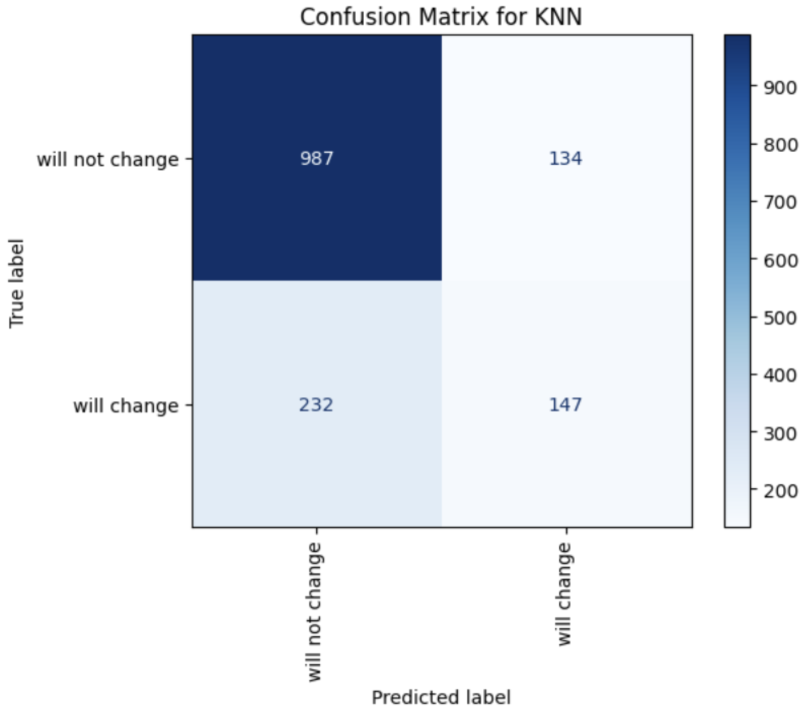
Each model was trained on the training set (X_train, y_train) and tested on the test set (X_test, y_test).

Comparison:

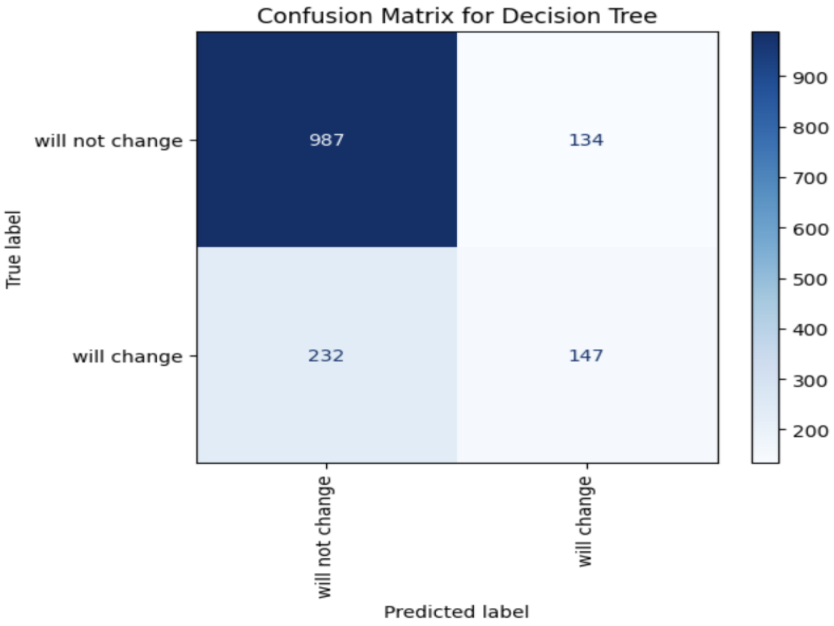
Model Name	Accuracy	Precision	Recall
KNN	74	0.49	0.33
Decision Tree	72	0.43	0.43
Logistic Regression	77	0.59	0.26
Neural Network	78	0.61	0.35

Confusion Matrix:

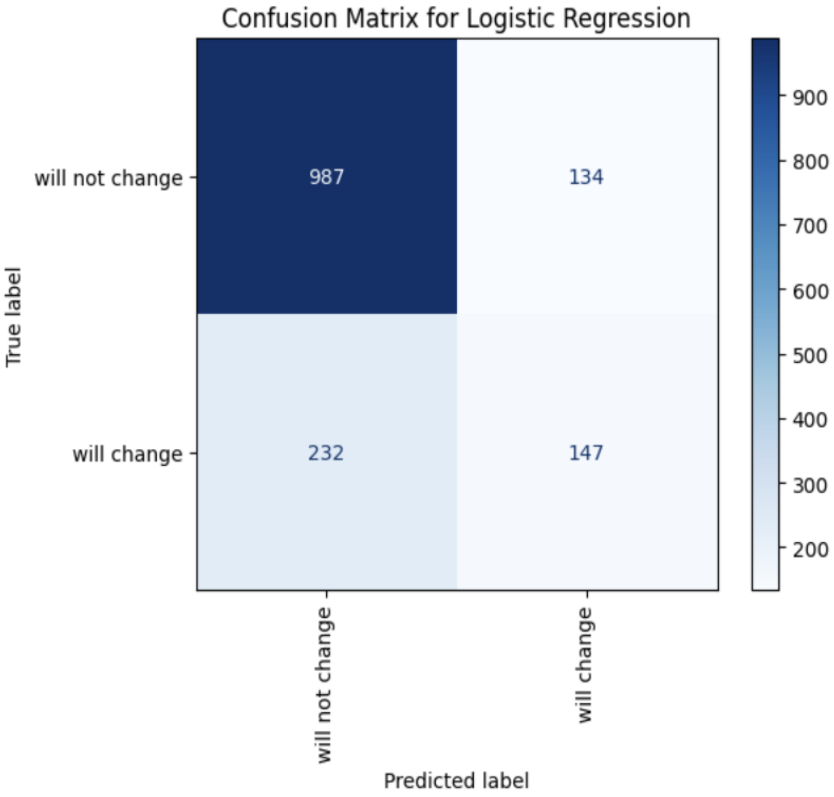
KNN



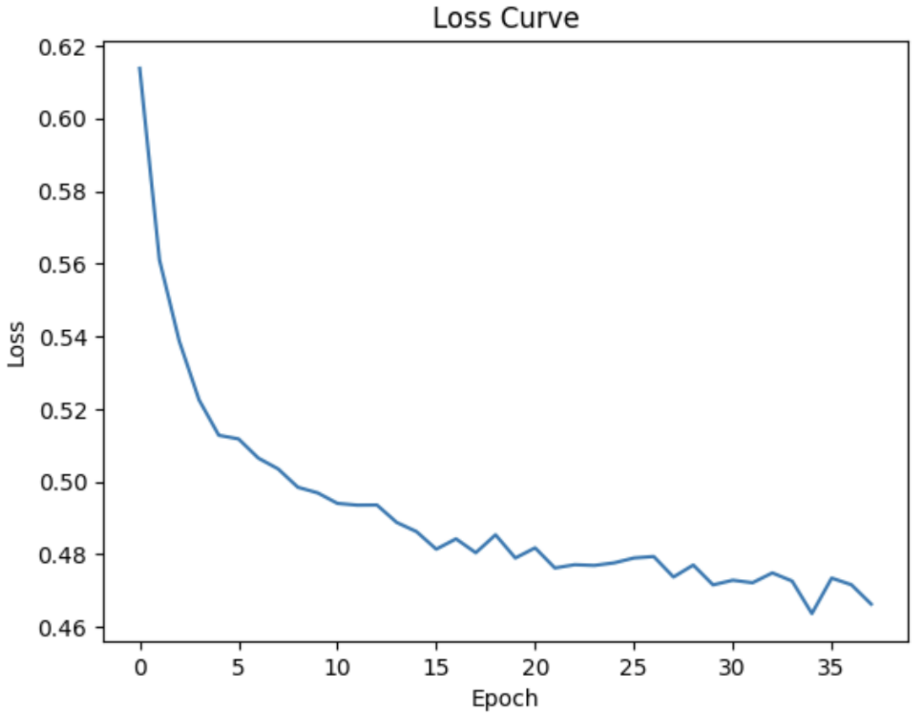
Decision Tree



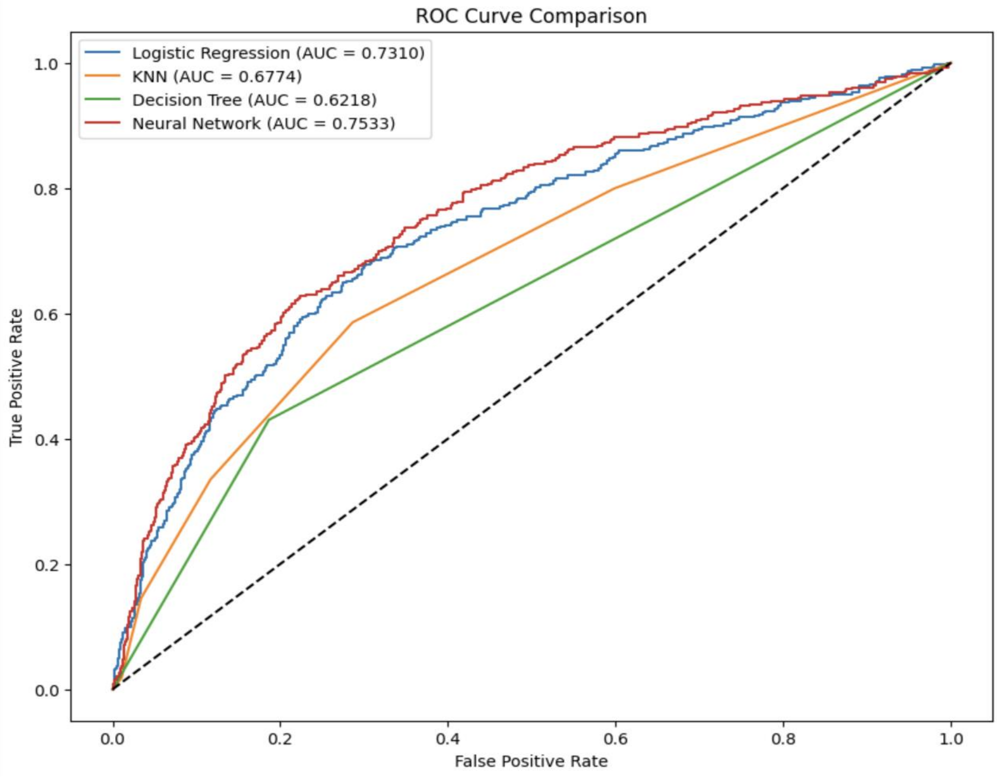
Logistic Regression:



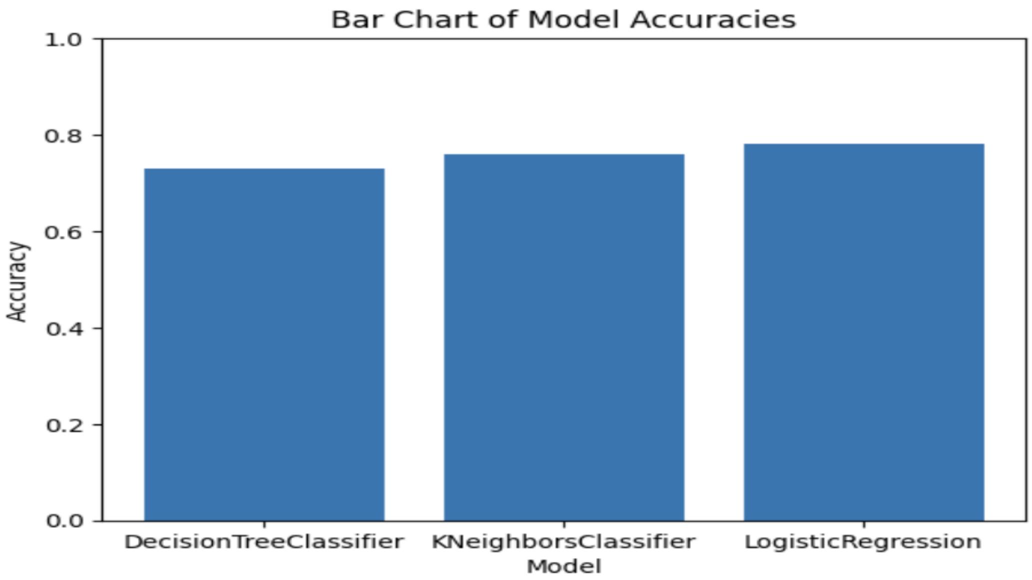
Loss Curve of Neural Network



AUC score, ROC curve



Accuracy Bar Chart:



8. Conclusion: After evaluating three different machine learning models — Decision Tree, K-Nearest Neighbors (KNN), and Logistic Regression — on the career switch prediction classification problem, the following accuracies were obtained: Decision Tree: 72%, K-Nearest Neighbors (KNN): 74%, and Logistic Regression: 77% and Neural Networks: 78%.

Among these models, Logistic Regression achieved the highest accuracy and proved to be the most effective for this prediction task. However, these results are subject to change with different datasets, and further improvement is possible through hyperparameter tuning, feature engineering, and testing with additional algorithms.

Reference:

TensorFlow. (n.d.). *Getting started with Keras & Sequential model.* TensorFlow. https://www.tensorflow.org/guide/keras/sequential_model

Scikit-learn developers. (n.d.). *User guide: Classification*. Scikit-learn. https://scikit-learn.org/stable/supervised_learning.html

Towards Data Science. (2020, July 20). *Understanding ROC Curves and AUC in Machine Learning.* Medium. https://towardsdatascience.com/understanding-roc-curves-and-auc-840b6c2839b7

GeeksforGeeks. (n.d.). *Logistic Regression in Machine Learning*. GeeksforGeeks. https://www.geeksforgeeks.org/logistic-regression/

GeeksforGeeks. (n.d.). *K-Nearest Neighbors (KNN) Algorithm in Python.* GeeksforGeeks. https://www.geeksforgeeks.org/k-nearest-neighbours/